

Bisma Farhat

39967

Lab 6

```
import random

class Environment:
    def __init__(self):

        self.locationCondition = {
            'A': random.randint(0, 1),
            'B': random.randint(0, 1),
            'C': random.randint(0, 1),
            'D': random.randint(0, 1)
        }

class SimpleReflexVacuumAgent:
    def __init__(self, environment):
        self.environment = environment
        self.score = 0

        self.vacuumLocation = random.choice(['A', 'B', 'C', 'D'])

        print("Initial environment conditions:",
self.environment.locationCondition)

        print(f"Vacuum is randomly placed at Location
{self.vacuumLocation}.")
        self.clean_location(self.vacuumLocation)

        locations = ['A', 'B', 'C', 'D']
        for location in locations:
            if location != self.vacuumLocation:
                self.move_and_clean(location)

        print("Final environment conditions:",
self.environment.locationCondition)
        print("Performance Measurement: " + str(self.score))

    def clean_location(self, location):
```

Bisma Farhat

39967

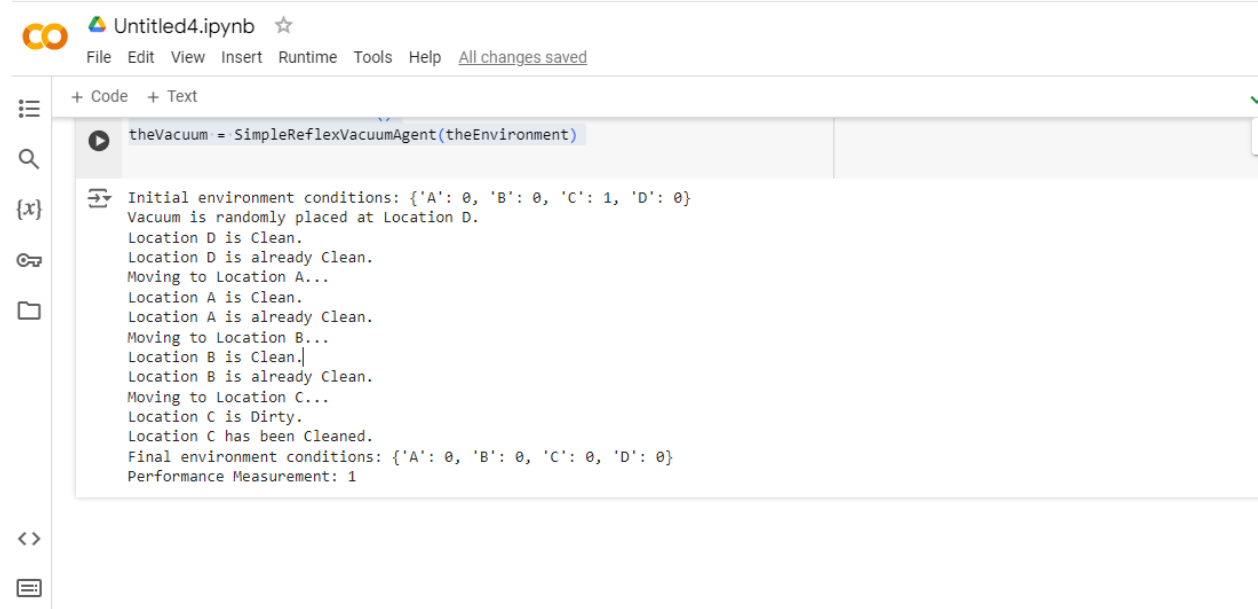
Lab 6

```
        print(f"Location {location} is {'Dirty' if
self.environment.locationCondition[location] == 1 else 'Clean'}.")
        if self.environment.locationCondition[location] == 1:
            self.environment.locationCondition[location] = 0
            self.score += 1
            print(f"Location {location} has been Cleaned.")
        else:
            print(f"Location {location} is already Clean.")

def move_and_clean(self, location):
    print(f"Moving to Location {location}...")
    self.clean_location(location)
```

```
theEnvironment = Environment()
theVacuum = SimpleReflexVacuumAgent(theEnvironment)
```

Output:



The screenshot shows a Jupyter Notebook titled "Untitled4.ipynb". The code cell contains the line `theVacuum = SimpleReflexVacuumAgent(theEnvironment)`. The output of this cell is a series of messages from the vacuum agent, detailing its initial conditions, random placement at Location D, and its subsequent actions of cleaning locations D, A, B, and C. The final output shows the environment conditions as all clean and a performance measurement of 1.

```
theVacuum = SimpleReflexVacuumAgent(theEnvironment)
```

Initial environment conditions: {'A': 0, 'B': 0, 'C': 1, 'D': 0}
Vacuum is randomly placed at Location D.
Location D is Clean.
Location D is already Clean.
Moving to Location A...
Location A is Clean.
Location A is already Clean.
Moving to Location B...
Location B is Clean.
Location B is already Clean.
Moving to Location C...
Location C is Dirty.
Location C has been Cleaned.
Final environment conditions: {'A': 0, 'B': 0, 'C': 0, 'D': 0}
Performance Measurement: 1